The Patent Office

INVESTOR IN PEOPLE

The Patent Office
Concept House
Cardiff Road
Newport
South Wales
NP10 8QQ

I, the undersigned, being an officer duly authorised in accordance with Section 74(1) and (4) of the Deregulation & Contracting Out Act 1994, to sign and issue certificates on behalf of the Comptroller-General, hereby certify that annexed hereto is a true copy of the documents as originally filed in connection with the patent application identified therein.

In accordance with the Patents (Companies Re-registration) Rules 1982, if a company named in this certificate and any accompanying documents has re-registered under the Companies Act 1980 with the same name as that with which it was registered immediately before re-registration save for the substitution as, or inclusion as, the last part of the name of the words "public limited company" or their equivalents in Welsh, references to the name of the company in this certificate and any accompanying documents shall be treated as references to the name with which it is so re-registered.

In accordance with the rules, the words "public limited company" may be replaced by p.l.c., plc, P.L.C. or PLC.

Re-registration under the Companies Act does not constitute a new legal entity but merely subjects the company to certain additional company law rules.

Signed M Cunkin

Dated 6-3-00

CERTIFIED COPY OF
PRIORITY DOCUMENT

Patents Act 1977
(Rule 16)

23 APR 1997
RECEIVED BY HAND

Office

24APR97 E267036-1 001822
P01/7700 25.00

Request for grant of a patent

(See the notes on the back of this form. You can also get an explanatory leaflet from the Patent Office to help you fill in this form)

The Patent Office

Cardiff Road
Newport
Gwent NP9 1RH

1. Your reference

R/97005Q2/JDR

2. Patent application number
(The Patent Office will fill in this part)

9708172.3 23 APR 1997

3. Full name, address and postcode of the or of each applicant *(underline all surnames)*

Xerox Corporation

Xerox Square
Rochester
New York 14644
United States of America

Patents ADP number *(if you know it)*

39946185001

If the applicant is a corporate body, give the country/state of its incorporation

New York, USA

4. Title of the invention

KNOWLEDGE BROKERS USING SIGNED FEATURE CONSTRAINTS

5. Name of your agent *(if you have one)*

Julian D. Reynolds

"Address for service" in the United Kingdom to which all correspondence should be sent *(including the postcode)*

Rank Xerox Limited
Patent Department
Parkway
Marlow
Bucks SL7 1YL
United Kingdom

Patents ADP number *(if you know it)*

6329718002

6. If you are declaring priority from one or more earlier patent applications, give the country and the date of filing of the or of each of these earlier applications and *(if you know it)* the or each application number

Country	Priority application number *(if you know it)*	Date of filing *(day / month / year)*

7. If this application is divided or otherwise derived from an earlier UK application, give the number and the filing date of the earlier application

Number of earlier application	Date of filing *(day / month / year)*

8. Is a statement of inventorship and of right to grant of a patent required in support of this request? *(Answer 'Yes' if:*
a) any applicant named in part 3 is not an inventor, or
b) there is an inventor who is not named as an applicant, or
c) any named applicant is a corporate body.
See note (d))

Yes

9. Enter the number of sheets for any of the following items you are filing with this form. Do not count copies of the same document

Continuation sheets of this form	
Description	12
Claim(s)	4
Abstract	1
Drawing(s)	9

10. If you are also filing any of the following, state how many against each item.

Priority documents	
Translations of priority documents	
Statement of inventorship and right to grant of a patent *(Patents Form 7/77)*	1
Request for preliminary examination and search *(Patents Form 9/77)*	1
Request for substantive examination *(Patents Form 10/77)*	
Any other documents *(please specify)*	

11. I/We request the grant of a patent on the basis of this application.

Signature Julian D. Reynolds Date 11.4.97

12. Name and daytime telephone number of person to contact in the United Kingdom

Julian D. Reynolds (01628 893447)

Warning

After an application for a patent has been filed, the Comptroller of the Patent Office will consider whether publication or communication of the invention should be prohibited or restricted under Section 22 of the Patents Act 1977. You will be informed if it is necessary to prohibit or restrict your invention in this way. Furthermore, if you live in the United Kingdom, Section 23 of the Patents Act 1977 stops you from applying for a patent abroad without first getting written permission from the Patent Office unless an application has been filed at least 6 weeks beforehand in the United Kingdom for a patent for the same invention and either no direction prohibiting publication or communication has been given, or any such direction has been revoked.

Notes

a) *If you need help to fill in this form or you have any questions, please contact the Patent Office on 0645 500505.*

b) *Write your answers in capital letters using black ink or you may type them.*

c) *If there is not enough space for all the relevant details on any part of this form, please continue on a separate sheet of paper and write "see continuation sheet" in the relevant part(s). Any continuation sheet should be attached to this form.*

d) *If you have answered 'Yes' Patents Form 7/77 will need to be filed.*

e) *Once you have filled in the form you must remember to sign and date it.*

f) *For details of the fee and ways to pay please contact the Patent Office.*

DUPLICATE

KNOWLEDGE BROKERS USING SIGNED FEATURE CONSTRAINTS

The present invention relates to data processing, and more particularly relates to the transfer between computing devices, and the retrieval by such devices, of information or knowledge using signed feature constraints.

With widespread availability of new electronic sources of information, such as e-mail, internet access and on-line information repositories, the number of electronic documents available to a computer user is multiplying. Documents can also be built dynamically by accessing and combining information existing over distributed sources. Hierarchical mark-up languages such as SGML can be used to define document templates that can be dynamically filled in with heterogeneous components. These documents can in turn be made permanent by storing them in document management systems, thus entering them in the normal document lifecycle.

Attempts at standardisation have led to the Document Management Alliance (DMA) industry standard, concerned with search, retrieval, storage and conversion of electronic documents on heterogeneous document management systems.

In implementing a system a knowledge brokerage system for carrying out search and retrieval in accordance with such a standard, knowledge brokers may be used. Brokers are software agents which can process knowledge search requests. Knowledge is taken here to be any piece of electronic information intended to be publicly accessible. Different, possibly distributed, information sources are assumed to be available, from a simple file in a user's directory to a database local to a site, up to a wide area information service (WAIS) on the internet, for example.

When receiving a request, a broker may have sufficient knowledge to process it, or may need to retrieve more knowledge. For that purpose, it releases sub-requests, aimed at other brokers. Thus, knowledge retrieval is achieved by the collaboration of all the brokers which are alternatively service providers processing requests and clients of these services generating sub-requests.

In order to collaborate, the brokers must at least understand each other. This means that all the requests must be formulated in a common language (and also all the answers to the requests), even if the brokers may perform local translations. Logic provides the adequate language for such a purpose. A request can be expressed by a pair $\langle x, P\rangle$ where x is a logical variable and P a logical formula involving x, meaning "*Retrieve* knowledge objects x such that the property expressed by formula P holds". Interestingly, an answer to such a request can be expressed in the same formalism, i.e. a pair $\langle x, Q\rangle$ meaning "*There exists* a knowledge object x satisfying the property expressed by formula Q". The requirement here is that P must be a logical consequence of Q, so that the answer contains at least as much knowledge as the request. Moreover, the same logical formalism can be used to capture the scope of a broker, i.e. the area of knowledge it is concerned with: a broker with scope $\langle x, R\rangle$ means "*I am not capable of retrieving* knowledge objects x which do not satisfy the property expressed by formula R". In many situations, the scope of a broker may vary, because it gets specialised or, on the contrary, expands its capacities, either externally or due to the knowledge retrieval process itself.

In other words, logic provides a common language where both requests, answers and scopes can be expressed. Brokers then perform logical operations on these three components. The most important logical operation, from which all the others can be reconstructed, is satisfiability checking, i.e. deciding whether some object could satisfy the property expressed by a formula, or, on the contrary, whether it is intrinsically contradictory. Unfortunately, it is well known that this operation, for *full* Classical Logic, is not algorithmic, i.e. it is provably impossible to write a program which implements it and always terminates. Given this limitation, a lot of research in knowledge representation has been focused on identifying *fragments* of Classical Logic in which satisfiability is algorithmically decidable. The trade-off here is between expressive power and tractability: the empty fragment, for example, is obviously tractable, but it is not very expressive.

The most popular fragment which emerged is known as "feature constraints" (FCs). The satisfiability problem in this case is also known as "feature constraint solving".

As is known, feature constraints are built from atomic constraints which are either sorts or features. A sort is a unary relation, expressing a property of a single entity. For example, `P:person` expresses that an entity `P` is of sort `person`. A feature is a binary relation expressing a property linking two entities. For example, `P:employer->E` expresses that entity `P` has an employer, which is an entity `E`. Apart from sorts and features, most feature systems also allow built-in relations such as equality and disequality.

The full fragment of feature constraints, where the atomic components mentioned above are allowed to be combined by all the logical connectives (conjunction, disjunction, negation and quantifiers), although very expressive, is hardly tractable. A subfragment called "basic feature constraints" (BFC) has been considered, where negation and disjunction are simply forbidden. Efficient constraint solving algorithms have been proposed for this sub-fragment. However, a drawback is that the complete absence of negation puts strong limitations on the kind of operations a knowledge broker may wish to perform.

There is therefore a need for techniques which avoid the abovementioned problems and provide tractable solutions. It would be desirable to have a system which provided brokers with a more powerful set of available operations.

The present invention provides a method carried out in a data processing device including a processor, memory, and a user interface, the data processing device being couple in a network to one or more other data processing devices, at least one of the data processing devices including means for storing a repository of electronic documents, comprising: (a) receiving at least one user input designating a feature constraint, said feature constraint comprising at least a positive component and a negative component, each of the positive component and the negative component including one or more relations, the or each relation defining a document related entity and a property of the entity; (b) solving the feature constraint to determine from the positive and negative components one or more document references, the or each document reference corresponding to a document within said repository satisfying said feature constraint.

The invention further provides a method carried out in a data processing device including a processor, memory, and a user interface, comprising: (i) receiving a first user input designating a graphical

object corresponding to a stored feature constraint, said feature constraint comprising at least a positive component and a negative component, each of the positive component and the negative component including one or more relations, the or each relation defining a document related entity and a property of the entity, (j) receiving a second user input indicating that the feature constraint is to be sent to another data processing device, (k) encoding the feature constraint in a data packet, and (l) transmitting the data packet.

In each case, the method may include retrieving knowledge from a repository stored in a data processing device at each of a plurality of locations, and optionally combining each piece of knowledge so obtained to generate a new document.

The invention further provides a data processing device when suitably programmed for carrying out the methods as set forth above, or according to any of the appended claims, the device comprising a processor, a memory, and a user interface.

The invention further provides a data processing device comprising: a processor, a memory coupled to the processor, and a user interface coupled to the processor and to the memory and adapted to be operable by a user to generate user inputs, the data processing device being couple in a network to one or more other data processing devices, at least one of the data processing devices including means for storing a repository of electronic documents, the data processing device further comprising means for receiving at least one user input designating a feature constraint, said feature constraint comprising at least a positive component and a negative component, each of the positive component and the negative component including one or more relations, the or each relation defining a document related entity and a property of the entity, means for solving the feature constraint to determine from the positive and negative components one or more document references, the or each document reference corresponding to a document within said repository satisfying said feature constraint.

The invention further provides a system for accessing or distributing electronic documents, according to claim 19 of the appended claims.

The invention further provides a portable device for accessing or distributing electronic documents, according to claim 20 of the appended claims.

The invention further provides a apparatus for scanning, copying and/or printing documents, according to claim 21 of the appended claims.

The invention employs a subset of feature constraints — "signed feature constraints" (SFC) — and a method for solving SFCs. An advantage is that SFCs can be used in knowledge retrieval engines to specify, in a common language, (i) knowledge search requests, (ii) the answers to these requests and (iii) the state of the knowledge retrieving agents (referred to herein as knowledge brokers).

The infra-structure required to support collaboration, and the way knowledge is stored locally within each broker, may be in accordance with the model disclosed in Andreoli et al (1996), The Constraint-Based Knowledge Broker Model: Semantics, Implementation and Analysis, *J. Symbolic Computation*). The following discussion addresses rather the knowledge manipulations occurring within each broker.

Embodiments of the invention will now be described, by way of example, with reference to the accompanying drawings, in which:

Figure 1 illustrates schematically a data processing network which may be used to implement an embodiment of the invention;

Figure 2 illustrates schematically the scope defined by a signed feature constraint;

Figure 3 is a view of the user interface of a fixed computing device at one instant during the entry by a user of a query;

Figure 4 shows a schematic flow chart of the steps in entering elements of a query using the interface of Fig. 3;

Figure 5 illustrates a paper form suitable for use by a user in an alternative embodiment of the invention, for entering a query;

Figure 6 shows a schematic flow chart of the steps in entering elements of a query using the paper form of Fig. 5;

Figure 7 is a schematic flow chart of the steps in using a feature constraint to retrieve document references and display or print corresponding documents;

Figure 8 illustrates a portion of a list of hits obtained during the process of Fig. 7;

Figure 9 shows selected hits from the list of Fig. 8 after transformation into HTML format; and

Figure 10 illustrates a more detailed presentation of a single selected hit.

1. System hardware

It will be appreciated that the present invention may be implemented using conventional computer network technology, either using a local area network (LAN) or, more suitably, a wide area network (WAN). The invention has been implemented using conventional web browser software (e.g. Netscape) providing cross-platform communication and document transfer over the internet. This is schematically illustrated in Fig. 1. It will be appreciated that each machine 2, 4, 6 forming part of the network 21 may be a PC running Windows™, a Mac running MacOS, or a minicomputer running UNIX, which are well known in the art. For example, the PC hardware configuration is discussed in detail in *The Art of Electronics*, 2nd Edn, Ch. 10, P. Horowitz and W. Hill, Cambridge University Press, 1989. However, it will be appreciated that the invention may be implemented using different system configurations: see EP-A-691,619 (hereafter "EP'619".

Upon request of a user at machine 22 a document stored on machine 26 may be retrieved and sent from machine 26 over the internet, via any number of intermediate machine 24 to machine 22. As is well known, the document may be retrieved using as a unique identifier its World Wide Web URL, as discussed in EP'619. Preferably also connected to the network 21 are any number of printers or multifunction devices (capable of scanning/printing/faxing, etc.) (not shown), as discussed in EP'619. Multifunction devices are discussed in more detail in EP-A-741,487. Each machine coupled to the network may be equipped with appropriate hardware and software, which is known in the art, for communication with portable computing devices, such as personal digital assistants (PDAs), handheld PCs, or pocket or wristwatch computers. In this way, the requesting machine may generate a request in response to

receiving a data packet from a user of a portable computing device, as discussed in further detail in British patent application 97_____ (agent's reference R/97005Q1), filed concurrently herewith.

2. Principles of feature constraints

As mentioned above, the full fragment of feature constraints, where the atomic components mentioned above are allowed to be combined by all the logical connectives (conjunction, disjunction, negation and quantifiers), although very expressive, is hardly tractable. A subfragment called "basic feature constraints" (BFC) has been considered, where negation and disjunction are simply forbidden. Efficient constraint solving algorithms have been proposed for this sub-fragment. However, a drawback is that the complete absence of negation puts strong limitations on the kind of operations a knowledge broker may wish to perform.

In preferred embodiments, the present invention makes use of a powerful operation, referred to as "scope-splitting", which relies on the use of negation. Indeed, a broker may wish to split its scope, specified by a pair $\langle x, P \rangle$ according to a criterion expressed by a formula F, thus creating two brokers with scope $P \wedge F$ and $P \wedge \neg F$. Thus, a broker in charge of bibliographic information may wish to split its scope into two new scopes: "books written after 1950", which can be represented by the BFC

```
X
   X : book
   X : year -> Y
   Y > 1950
```

and its complement, i.e. "books written before 1950 *or* documents which are *not* books"; this latter scope cannot be expressed using BFC, because negation and disjunction cannot be dispensed with. It has been discovered that the scope splitting operation is useful in many situations, for example to implement brokers capable of memorising and re-using information gathered during their lifetime. Embodiments of the present invention make use of, on the one hand, a fragment of feature constraints, called "signed feature constraints" (SFC), which allows limited use of negation, precisely capable of expressing the kind of split scope mentioned above, and on the other hand, an efficient constraint solving method for SFC.

2.1 Signed Feature Constraints

A signed feature constraint is composed of a positive part and a list of negative parts, both of them being basic feature constraints. For example, the following signed feature constraint

```
P
+ P : person,
  P : employer-> E,
  E : "Xerox"
- P : nationality-> N,
  N : "American"
- P : spouse-> P'
  P': person
```

```
P': employer-> E'
E': "Xerox"
```

specifies a Xerox employee who is not American and is not married to another Xerox employee. This is represent it graphically as in Fig. 2. The round boxes denote the entities (logical variables), the sort relations (unary) are represented by dashed arrows labelled by the name of the sort in a square box, the feature relations (binary) are represented by plain arrows labelled by the name of the feature in a square box. The built-in predicates (not present in the example) are represented by rhombuses. The positive part of the SFC is contained in the top box and marks the distinguished entity of the scope (P in the example) by a double round box. The negative parts of the SFC are contained in the lower boxes in grey.

The main interest of SFC comes from the following property:

If the scope of a broker is represented by an SFC e_o, and this scope is split by a BFC e, then the two resulting split scopes e^+, e^- are both SFC.

Indeed, e^+ is obtained by merging the positive part of e_o with the BFC e; and e^- is obtained by extending e_o with a new negative part containing e alone. For example, assume a broker in charge of a bibliographical database containing various documents (books, videos etc.) about Art, but not authored by an American. It is represented by the SFC

```
X
+X : topic-> T
 T : "Art"
-X : author-> A
 A : nationality-> N
 N : "American"
```

It may be split by the constraint "books written after 1950", expressed by the BFC

```
X
 X : book
 X : year-> Y
 Y > 1950
```

The resulting scopes are simply

```
X
+X : book
 X : topic-> T
 X : year-> Y
 T : "Art"
 Y > 1950
-X : author-> A
 A : nationality-> N
 N : "American"
```

i.e. "Art books written after 1950 but not by an American author" and

```
X
```

```
+X : topic-> T
 T : "Art"
-X : author-> A
 A : nationality-> N
 N : "American"
-X : book
 X : year-> Y
 Y > 1950
```

i.e. "Art documents not authored by an American but not books subsequent to 1950".

2.2 Solving Signed Feature Constraints

Most constraint systems make a number of assumptions on the nature of sorts and features, called the axioms of the systems. These axioms are crucial to the satisfiability algorithm, since they determine when a feature constraint is contradictory and when it is satisfiable.

2.2.1 Feature Axioms

For the purpose of simplicity, the embodiment disclosed here makes use of a slight variant of the basic axiom system used in Aït-Kaci H. *et al.* (1994), A Feature-Based Constraint-System for Logic Programming with Entailment, *Theoretical Computer Science* **122**, pp. 263–283, although it will be appreciated by persons skilled in the art that the principles of the method apply to other sets of axioms as well.

1. Features are functional: this means that if two pairs of entities which are constrained by the same feature have the same first term, they also have the same second term. For example, it can be considered that the feature `spouse` is functional (within a specific cultural setting), meaning that a person cannot have two spouses: if, for a person `X`, we have `X:spouse->Y` and `X:spouse->Z`, then the entities `Y` and `Z` coincide (i.e. denote the same person). Other systems allow multi-valued features.

2. Sorts are disjoint: this means that no entity can be of two distinct sorts. For example, a book is not a person: we cannot have an entity `X` with `X:book` and `X:person`. Other systems consider hierarchies of sorts where some entities can have multiple sorts as long as they have a common denominator in the hierarchy.

3. There is a distinguished subset of sorts, called "value" sorts, so that no two distinct entities can be of the same value sort. Traditional basic elements (strings, numbers, etc.) are typical value sorts: for example, the string `"Xerox"` or the number `1950` are value sorts. Value sorts are not allowed to have features: this is the only axiom connecting sorts and features. Other systems consider more refined connections between sorts and features.

4. There is a distinguished built-in binary predicate, equality, with the traditional congruence axioms (which involve sorts and features). The axioms describing all the other built-in predicates are assumed to contain no mention of sorts and features.

These axioms are formally written in section A: Axioms in the Appendix at the end of this disclosure. They form a theory *T*.

2.2.2 Constraint Satisfaction

First, it is assumed that satisfiability over built-in predicates is decidable. This means that there is an algorithm which, given a formula *F* using only built-in predicates (*F* is also called a built-in constraint), can decide whether *F* is a logical consequence of the theory *T* (written $\vdash_T F$).

Constraint satisfaction over BFCs is defined by a set of conditional rewrite rules over BFCs (section B.1 of the Appendix) which have the following properties

(a) *The system of rules is convergent and hence defines a "normal form" for BFCs.* This can be shown in a classical way by proving that the system is "Church-Rosser" (critical pairs converge) and "Noetherian" (the size of the terms strictly decrease by rewriting).

(b) *A BFC is satisfiable if and only if its normal form is not reduced to the contradiction.* One implication can be proved by showing that rewrite steps preserve satisfiability. The reverse implication can be proved by displaying a model which satisfies BFCs whose normal form is not reduced to the contradiction.

Thus the rewrite rules describe the steps of the constraint satisfaction algorithm. This algorithm always terminates because the system of rewrite rules is convergent. It is to be noted that the definition of the rules rely on satisfiability tests of built-in constraints, which has been assumed decidable. This means that the algorithm is modular and can accommodate any kind of built-in constraints as long as a proper built-in constraint satisfaction algorithm is provided.

Using rewrite rules for constraint satisfaction algorithm can be implemented in a naive way in some symbolic language like Lisp or Prolog, or can be optimised, taking into account the properties of the specific built-in constraints which are used.

The algorithm for constraint satisfaction over SFCs (section B.2 of the Appendix) can informally be described as follows. Given an SFC, its positive component is first normalised by the algorithm for BFCs. If the result is a contradiction, the whole SFC is unsatisfiable. Otherwise, the positive component (normalised) is inserted in each of the negative components, which are then normalised by the algorithm for BFCs. If a resulting negative component has a contradictory normal form, it is eliminated, and if it has a tautological normal form the whole SFC is unsatisfiable. The normal form for SFCs thus obtained has the following property:

An SFC is satisfiable if and only if its normal form is not reduced to the contradiction. As in the previous case, the difficult part of the implication can be proved using model theory.

3. User transactions with feature constraints

Figure 3 is a view of the user interface of a fixed computing device at one instant during the entry by a user of a query. Within a dedicated window a main query entry box 30 is displayed, in a form well known in the art. For example, the user may be interested in entering a query along the lines 'books or articles after 1990 in which the title contains "constraints" but does not contain "internet"'. The query

box 30 includes boxes 31, 32 which the user can select by mouse inputs, and can use to type in, or complete elements (e.g. "books/articles") of the query. Buttons 33 may be used to select document-related entities, such as "title" and a constraint applying to it, such as "contains not". Additional buttons 34, 36 allow the user to restart, add to, edit and build up a query. Each element of the query is gradually added to the current specification of the query, which is displayed in it's current state in box 37. When the specification of the query is completed, button 38 is pressed to launch the search.

Figure 4 shows a schematic flow chart of the steps in entering elements (e.g. date after 90) of a query using the interface of Fig. 3. Initially, the knowledge broker main query window is displayed (step s41). The elements of the query are then received in turn as they are keyed in by the user (s42). For each query element confirmed by the user, the "current specification" is updated to include it any displayed (s43). On completion of the specification of the query, and selection by the user of the "Submit" button in Fig. 3, each query element is converted (s44) to its corresponding logical relation(s) - see section 2 above. The feature constraint is then compiled (s45) from the set of logical relations.

Figure 5 illustrates a paper form 50 suitable for use for entering a query by a user in an alternative embodiment of the invention. This embodiment is suitable for the user of a multifunction device, or a user having a scanner coupled to a computer. The form 50 used has several sections 51, 52, 53, 54, enabling the user to enter information about the type of document, author's name, date, and topic; however, it will be appreciated that any number of sections may be used, for entering any kind of information that a user may expect to have about a document. In this case, next to each option indicated by human readable text is a box which, when checked by a user, enables the choice to be determined by machine reading, as is known in the art. Certain boxes (55, 56, 57) may be used to enter hand-written information. alternatively, the query may be entered on a sheet entirely in typewritten or hand-written form, with the content of the query being determined by OCR and, where necessary, handwriting recognition.

Figure 6 shows a schematic flow chart of the steps in entering elements of a query using the paper form of Fig. 5. First, the sheet is scanned and a bitmapped image data file corresponding to the content of the sheet is stored (step s60). Then, (s61) an analysis is made of the image data at the locations corresponding the boxes 55-58, either as to whether the box was checked, or to extract the information written in the box. Then, for each section 51-54, the specified query element is determined (s62). where necessary by applying handwriting recognition and OCR (s63). Each query element is then converted to the corresponding logical relation(s) - see section 2 above. The feature constraint is then compiled (s45) from the set of logical relations.

Figure 7 is a schematic flow chart of the steps in using a feature constraint to retrieve document references and display or print corresponding documents. This may be performed by a conventional computer device, or by a multifunction device or printer equipped with a user interface. Initially, a FC is received from a user, for example in a data packet from a user of a portable device, as illustrated in Fig. 8, or by input directly into the machine by a user operating a keyboard and mouse, or touch screen, as is well known in the art. Upon the instigation of the user by appropriate input, the FC is solved as described in section 3 above, and the resulting request in the appropriate form passed to the search engine (s92). The search request is used to search all available repositories for documents

satisfying the FC (s93); and if necessary, the request may be broken down into subrequests as discussed in more detail in Andreoli et al (1996), The Constraint-Based Knowledge Broker Model: Semantics, Implementation and Analysis, *J. Symbolic Computation*).

Once obtained by the search engine, a list of hits — of documents satisfying the FC — is displayed (s94), as shown in Fig. 8. Then, in response to appropriate user input, operations may be performed to display individual hits with expanded detail of the document, to convert the document information to HTML format, or to download the document from the reposiitory (s95).

Figure 9 shows selected hits from the list of Fig. 8 after transformation into HTML format. As can be seen, for each hit there is displayed further information, such as author name, http_url, information source and title. If desired, the user can view the document for hit 1 by mouse clicking on the http_url displayed. The document can then be printed, if needed (s96).

Figure 10 illustrates a more detailed presentation of a single selected hit, i.e. with a set of attributes of the document. It can be seen that against one or more of the attributes are displayed URLs providing links to further pages providing information related to those attributes.

Appendix

A Axioms of the System

There are three sets of axioms.

Specific axioms for features and sorts :
Let τ, τ' denote any sorts, and f denote any feature.

$$\begin{array}{ll}
\forall x, y, z & x \xrightarrow{f} y \wedge x \xrightarrow{f} z \supset y = z \\
\forall x & \neg(x : \tau \wedge x : \tau') \text{ if } \tau \neq \tau' \\
\forall x, y & x : \tau \wedge y : \tau \supset x = y \text{ if } \tau \text{ is a value sort} \\
\forall x, y & \neg(x : \tau \wedge x \xrightarrow{f} y) \text{ if } \tau \text{ is a value sort}
\end{array}$$

Congruence axioms for equality :
Let p denote any built-in predicate. The traditional congruence axioms are:

$$\begin{array}{ll}
\forall x & x = x \\
\forall x, y & x = y \supset y = x \\
\forall x, y, z & x = y \wedge y = z \supset x = z \\
\forall x, y & x : \tau \wedge x = y \supset y : \tau \\
\forall x, y, z & x \xrightarrow{f} y \wedge x = z \supset z \xrightarrow{f} y \\
\forall x, y, z & x \xrightarrow{f} y \wedge y = z \supset x \xrightarrow{f} z \\
\forall \vec{x}, y & p(\vec{x}) \wedge x_i = y \supset p(\vec{y})
\end{array}$$

where i is some index in the list of variable $\vec{x}$ and $\vec{y}$ is identical to $\vec{x}$ except that $y_i = y$.

Built-in predicate axioms :
They must not mention sorts and features. For example, disequality can be axiomatized by

$$\begin{array}{ll}
\forall x, y & x \neq y \vee x = y \\
\forall x & \neg(x \neq x)
\end{array}$$

Precedence constraints are axiomatized by

$$\begin{array}{ll}
\forall x & \neg(x < x) \\
\forall x, y, z & x < y \wedge y < z \supset x < z
\end{array}$$

The built-in predicates $>, \leq, \geq$ can then be defined from $<$ and equality.

B Constraint Satisfaction

B.1 The BFC case

We represent a BFC as a pair $\langle B \mid \Gamma \rangle$ where B is a built-in constraint and Γ an *unordered* list of sort and feature constraints (read conjunctively). $\perp$ denotes the contradiction.

There are two sets of rewrite rules. The following rules correspond to simplifications of the BFCs.

$$\begin{array}{lcl}
\langle B \mid x \xrightarrow{f} y, z \xrightarrow{f} t, \Gamma\rangle & \longmapsto & \langle B \wedge y = t \mid x \xrightarrow{f} y, \Gamma\rangle \text{ if } \vdash_{\mathcal{T}} B \supset x = z \\
\langle B \mid x : \tau, y : \tau, \Gamma\rangle & \longmapsto & \langle B \mid x : \tau, \Gamma\rangle \text{ if } \vdash_{\mathcal{T}} B \supset x = y \text{ and } \tau \text{ is not a value sort} \\
\langle B \mid x : \tau, y : \tau, \Gamma\rangle & \longmapsto & \langle B \wedge x = y \mid x : \tau, \Gamma\rangle \text{ if } \tau \text{ is a value sort}
\end{array}$$

The following rules correspond to the detection of inconsistencies.

$$\begin{array}{lcl}
\langle B \mid \Gamma\rangle & \longmapsto & \perp \text{ if } \vdash_{\mathcal{T}} \neg B \\
\langle B \mid x : \tau, y : \tau', \Gamma\rangle & \longmapsto & \perp \text{ if } \vdash_{\mathcal{T}} B \supset x = y \text{ and } \tau \neq \tau' \\
\langle B \mid x : \tau, y \xrightarrow{f} z, \Gamma\rangle & \longmapsto & \perp \text{ if } \vdash_{\mathcal{T}} B \supset x = y \text{ and } \tau \text{ is a value sort}
\end{array}$$

The following property justifies the algorithm

$$\langle B \mid \Gamma\rangle \longmapsto \perp \text{ if and only if } \vdash_{\mathcal{T}} \forall\neg(B \wedge \bigwedge_{c \in \Gamma} c)$$

B.2 The SFC case

We represent an SFC as an *unordered* list of BFCs prefixed with a sign (+ or −); by definition, one and only one component is positive. Let $\mathcal{S}$ be an SFC. The SFC-normal form of $\mathcal{S}$ is written $\mathcal{S}^*$ and is obtained by the following algorithm:

Let c_o be the BFC-normal form of the positive component of $\mathcal{S}$.
If $c_o = \perp$ **Then**
  Return $\perp$
Else
  c_o is of the form $\langle B_o \mid \Gamma_o\rangle$
  Let $\{\langle B_i \mid \Gamma_i\rangle\}_{i=1,\ldots,n}$ be the list of negative components of $\mathcal{S}$.
  Foreach $i = 1, \ldots, n$
    Let c_i be the BFC normal form of $\langle B_o \wedge B_i \mid \Gamma_o, \Gamma_i\rangle$.
  If there exists $i \in 1, \ldots, n$ such that $c_i = \langle B \mid \Gamma\rangle$ and $\vdash_{\mathcal{T}} B$ and Γ is empty **Then**
    Return $\perp$
  Else
    Let $I = \{i \in 1, \ldots, n \text{ such that } c_i \neq \perp\}$
    Return $\{+c_o, \{-c_i\}_{i \in I}\}$

The following property justifies the algorithm

$$[+\langle B_o \mid \Gamma_o\rangle, \{-\langle B_i \mid \Gamma_i\rangle\}_{i=1}^{n}]^* = \perp \text{ if and only if } \vdash_{\mathcal{T}} \forall\neg[(B_o \wedge \bigwedge_{c \in \Gamma_o} c) \wedge \bigwedge_{i=1}^{n} \neg(B_i \wedge \bigwedge_{c \in \Gamma_i} c)]$$

Claims:

1. A method carried out in a data processing device including a processor, memory, and a user interface, the data processing device being couple in a network to one or more other data processing devices, at least one of the data processing devices including means for storing a repository of electronic documents, comprising:

(a) receiving at least one user input designating a feature constraint, said feature constraint comprising at least a positive component and a negative component, each of the positive component and the negative component including one or more relations, the or each relation defining a document related entity and a property of the entity,

(b) solving the feature constraint to determine from the positive and negative components one or more document references, the or each document reference corresponding to a document within said repository satisfying said feature constraint.

2. The method of claim 1, further comprising the step of:

(c) displaying the or each document reference determined in step (b).

3. The method of claim 2, further comprising the step of:

(d) in response to a second user input designating one of the displayed document references, retrieving the document corresponding to said document reference from said repository and, optionally, displaying said document or a portion thereof.

4. The method of claim 2 or 3, further comprising the step of:

(e) in response to a third user input, causing the document corresponding to said document reference to be printed.

5. The method of any of the preceding claims, wherein step (a) comprises:

(a1) receiving at least one user input, the user input(s) defining at least one relation, the or each relation defining a document related entity and a property of the entity,

(a2) compiling a feature constraint, the feature constraint including the relation(s) received in step (a1).

6. The method of any of the preceding claims, wherein, prior to step (a2), step (a1) is repeated a plurality of times.

7. The method of claim 5 or 6, wherein the data processing device comprising device comprises a portable computing device, and step (a1) comprises receiving user inputs via touchscreen or a keyboard.

8. The method of claim 5 or 6, wherein the data processing device comprising device comprises a fixed computing device, and step (a1) comprises receiving user inputs via touchscreen, a keyboard, and/or mouse.

9. The method of claim 8, wherein the fixed computing device comprises a multifunction device including scanning means,

wherein step (a1) comprises:

(a1i) scanning an image-bearing portable medium to produce electronic signals corresponding to at least one predetermined portion of the medium; and

(a1ii) determining from said signals a relation corresponding thereto.

10. The method of claim 9, wherein said medium bears, for each said predetermined portion, a human readable indication of the disposition of said predetermined portion and, associated therewith, a human readable designator of the relation corresponding to said predetermined portion.

11. The method of any of claims 1 to 4, wherein step (a) comprises:

(f) receiving a data packet

(g) decoding the data packet to derive a feature constraint, the feature constraint comprising one or more relations, the or each relation defining a document related entity and a property of the entity,

(h) storing the feature constraint derived in step g).

12. A method carried out in a data processing device including a processor, memory, and a user interface, comprising:

(i) receiving a first user input designating a graphical object corresponding to a stored feature constraint, , said feature constraint comprising at least a positive component and a negative component, each of the positive component and the negative component including one or more relations, the or each relation defining a document related entity and a property of the entity,

(j) receiving a second user input indicating that the feature constraint is to be sent to another data processing device,

(k) encoding the feature constraint in a data packet, and

(l) transmitting the data packet.

13. The method of any of the preceding claims, wherein one of said relations is a sort relation, the sort relation indicating the sort of which the respective entity comprises.

14. The method of any of the preceding claims, wherein one of said relations is a feature relation, the feature relation indicating that the respective entity has an attribute, the value of said attribute being a second entity.

15. The method of any of the preceding claims, wherein the property is a value, or range of values, which said entity may have.

16. A data processing device when suitably programmed for carrying out the method of any of the preceding claims, the device comprising a processor, a memory, and a user interface.

17. A data processing device comprising:

a processor,

a memory coupled to the processor, and

a user interface coupled to the processor and to the memory and adapted to be operable by a user to generate user inputs,

the data processing device being couple in a network to one or more other data processing devices, at least one of the data processing devices including means for storing a repository of electronic documents, the data processing device further comprising

means for receiving at least one user input designating a feature constraint, said feature constraint comprising at least a positive component and a negative component, each of the positive component and the negative component including one or more relations, the or each relation defining a document related entity and a property of the entity,

means for solving the feature constraint to determine from the positive and negative components one or more document references, the or each document reference corresponding to a document within said repository satisfying said feature constraint.

18. The method of claim 17, wherein said means for receiving a feature constraint comprises

means for receiving at least one user input, the user input(s) defining at least one relation, and for compiling the feature constraint, the feature constraint including the received relations.

19. A system for accessing or distributing electronic documents, including:

means storing a repository of electronic documents, each document having a corresponding document reference, and

a plurality of objects, at least one of said objects being portable or mobile, each object including means for communicating with the or each other object and with a user interface, and means for receiving, storing and/or transmitting a feature constraint, said feature constraint comprising at least a positive component and a negative component, each of the positive component and the negative component including one or more relations, the or each relation defining a document related entity and a property of the entity.

20. A portable device for accessing or distributing electronic documents, including:

means for communicating with fixed or mobile electronic devices and with a user interface, at least one of said devices including means storing a repository of electronic documents, each document having a corresponding document reference, and

means for receiving, storing and/or transmitting a feature constraint, said feature constraint comprising at least a positive component and a negative component, each of the positive component and the negative component including one or more relations, the or each relation defining a document related entity and a property of the entity.

21. An apparatus for scanning, copying and/or printing documents, including:

means for accessing a repository of electronic documents, each electronic document having a corresponding document reference, means for communicating with one or more of a plurality of objects, at least one of said objects being portable or mobile, and with a user interface, and

means for receiving, storing and/or transmitting a feature constraint, said feature constraint comprising at least a positive component and a negative component, each of the positive component and the negative component including one or more relations, the or each relation defining a document related entity and a property of the entity.

ABSTRACT

KNOWLEDGE BROKERS USING SIGNED FEATURE CONSTRAINTS

A method of implementing a knowledge broker software agent which processes knowledge requests expressed in terms of feature constraints, if necessary decomposing the requests into subrequests. The invention employs a subset of feature constraints, called "signed feature constraints" (SFC), each feature constraint having at least a positive component and a negative component, each of the positive component and the negative component including one or more relations, such as sort relations and feature relations. An advantage is that SFCs can be used in knowledge retrieval engines to specify, in a common language, (i) knowledge search requests, (ii) the answers to these requests and (iii) the state of the knowledge retrieving agents (referred to herein as knowledge brokers). A method for solving SFCs is also disclosed.

[Fig. 3]

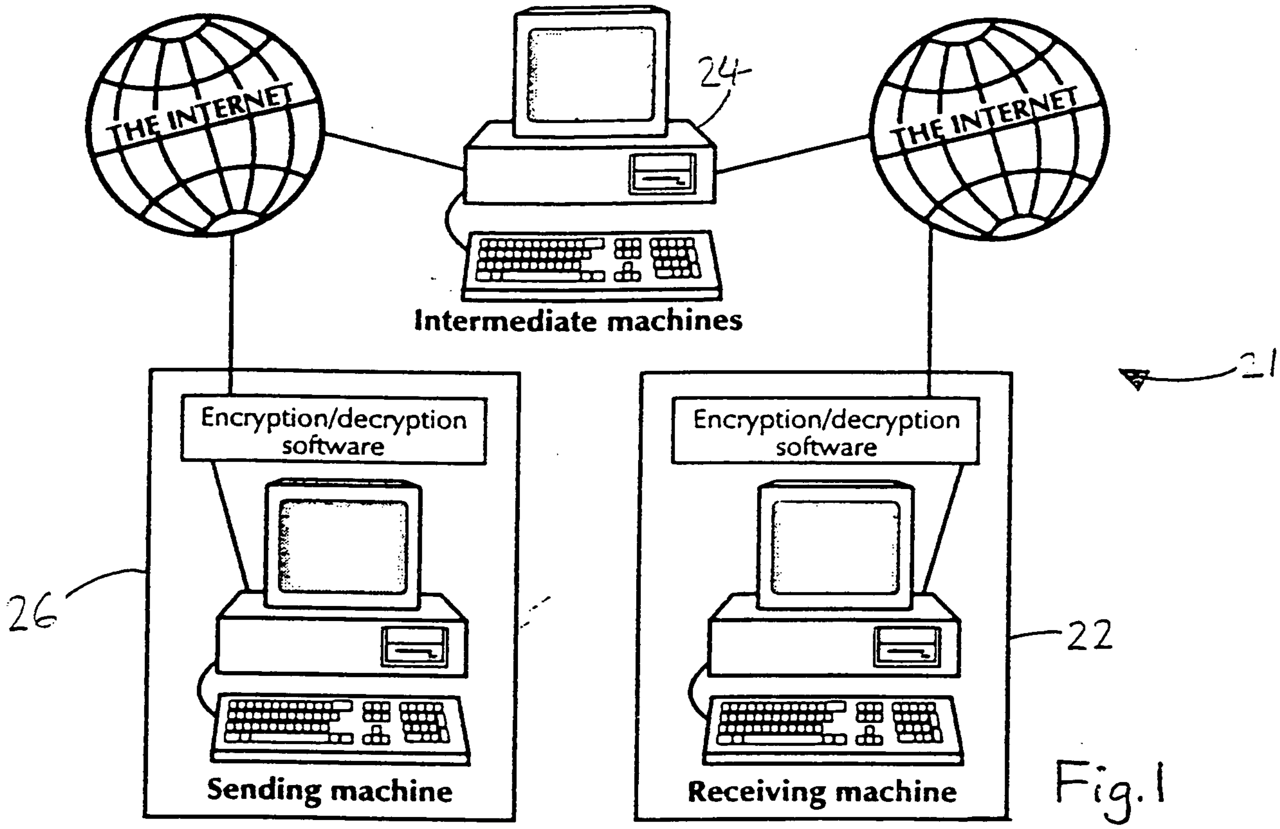

THE INTERNET
24
THE INTERNET
Intermediate machines
21
Encryption/decryption software
Encryption/decryption software
26
22
Sending machine
Receiving machine

Fig. 1

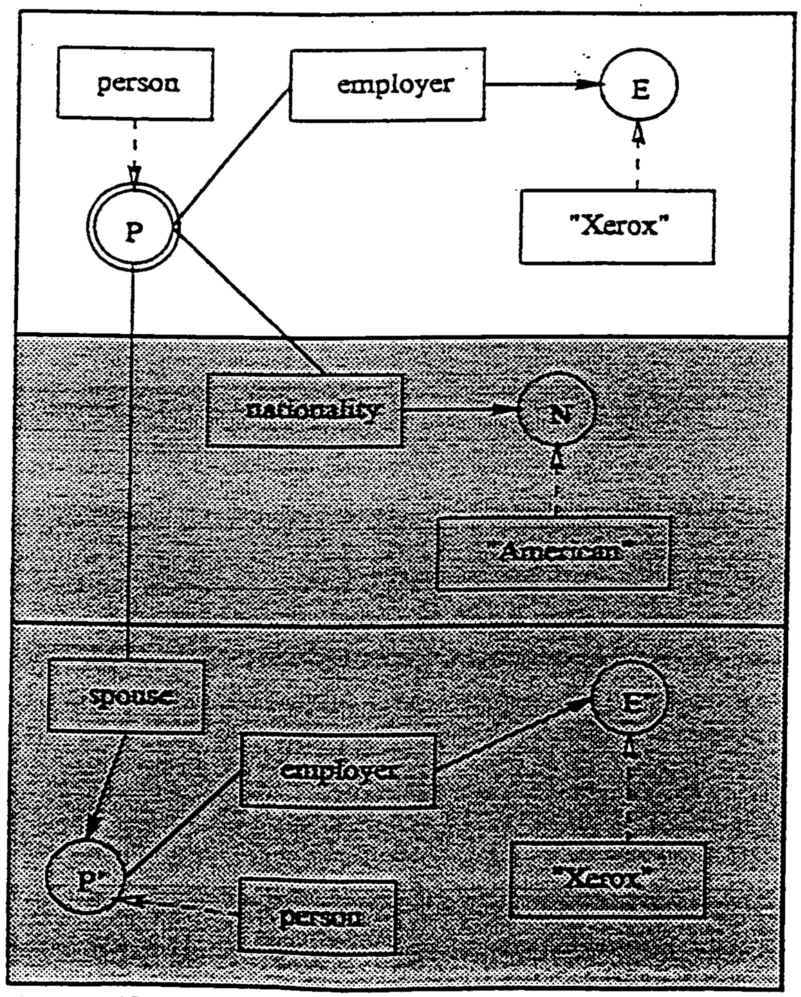

person
employer
E
P
"Xerox"
nationality
N
"American"
spouse
E'
employer
P'
person
"Xerox"

Fig. 2

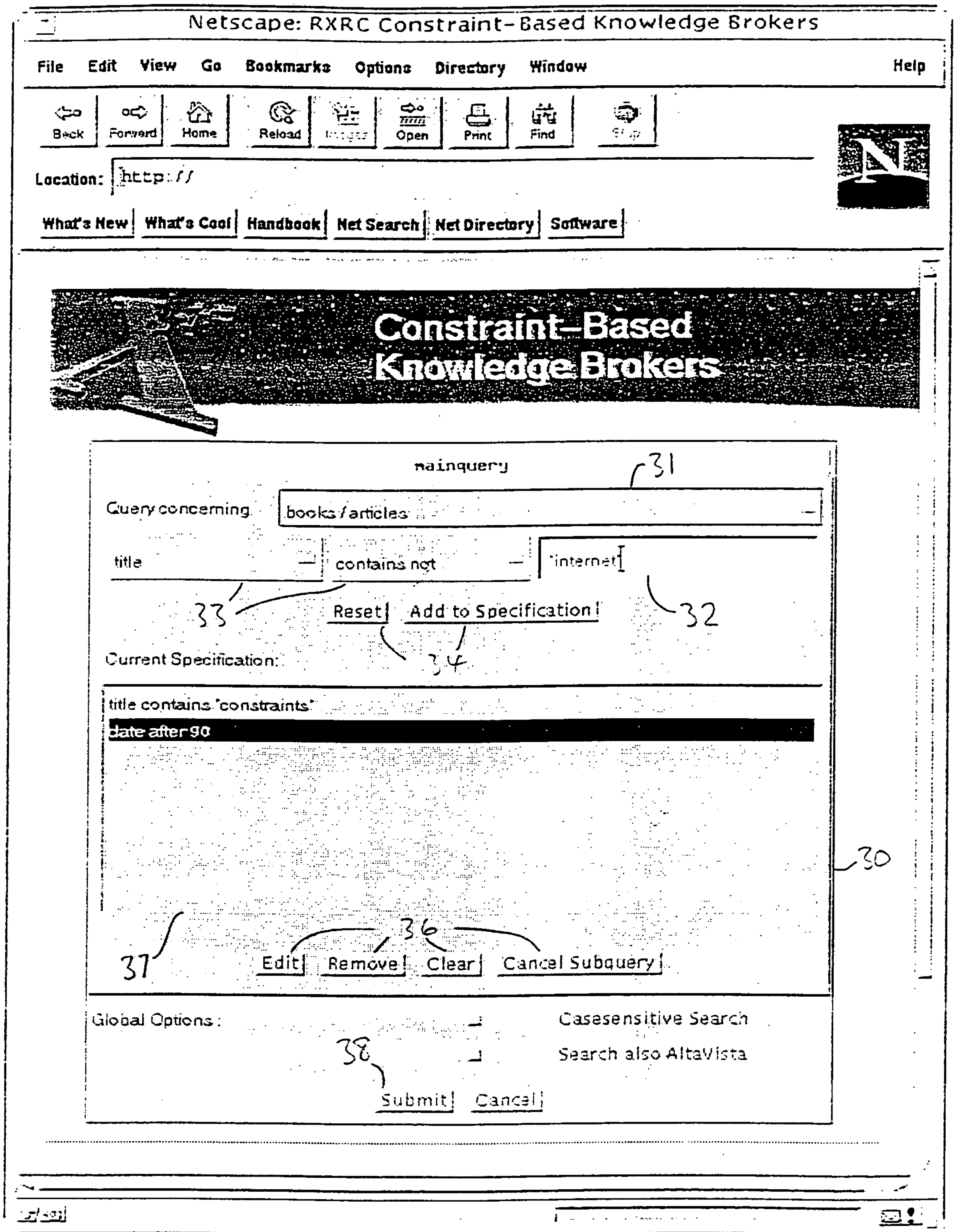

Netscape: RXRC Constraint-Based Knowledge Brokers
File Edit View Go Bookmarks Options Directory Window Help
Back Forward Home Reload Images Open Print Find Stop
Location: http://
What's New What's Cool Handbook Net Search Net Directory Software
Constraint-Based Knowledge Brokers
mainquery
31
Query concerning books/articles
title contains not "internet"
33 Reset Add to Specification 32
34
Current Specification:
title contains "constraints"
date after 90
30
37 36
Edit Remove Clear Cancel Subquery
Global Options: Casesensitive Search
38 Search also AltaVista
Submit Cancel

Fig. 3

SPARE

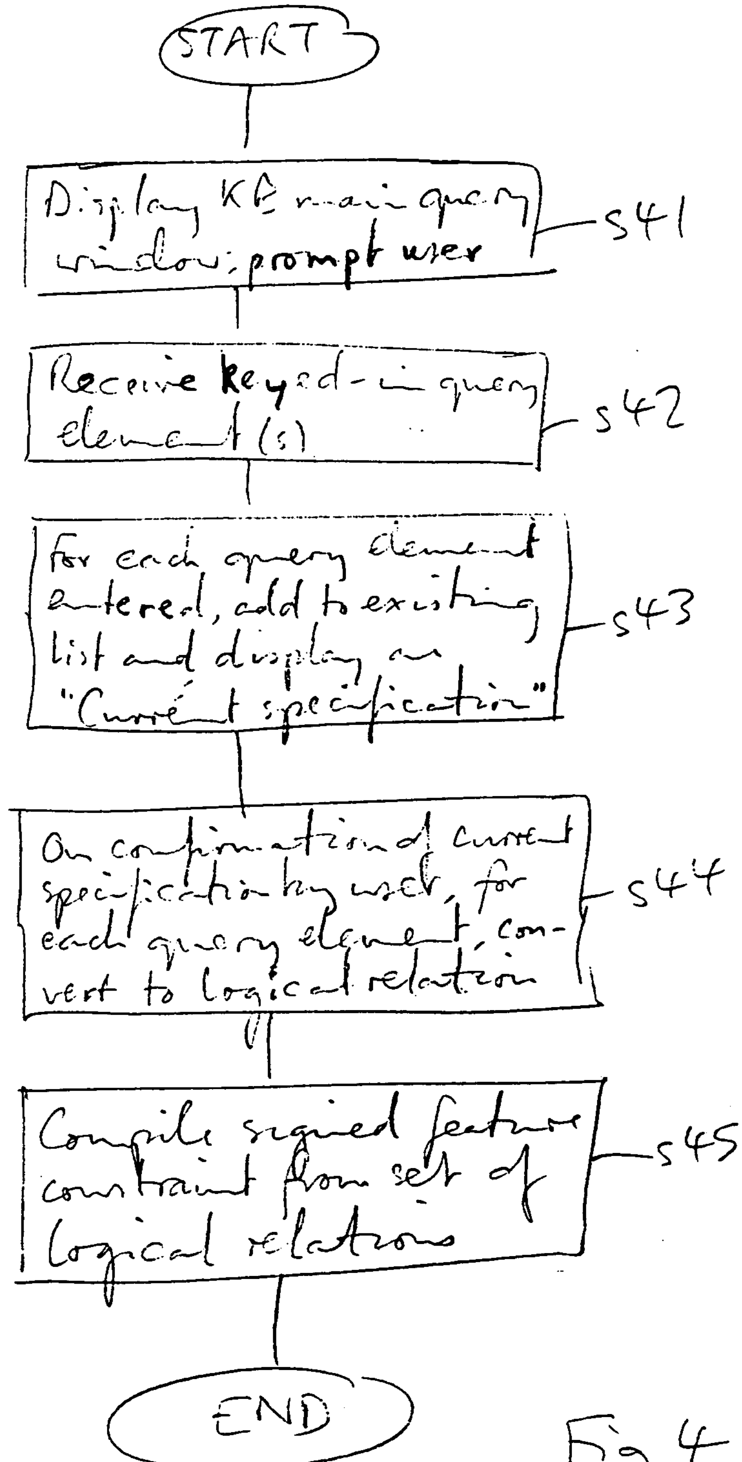

START
Display KB main query window; prompt user
S41
Receive keyed-in query element(s)
S42
For each query element entered, add to existing list and display as "Current specification"
S43
On confirmation of current specification by user, for each query element, convert to logical relation
S44
Compile signed feature constraint from set of logical relations
S45
END

Fig. 4

SPARE

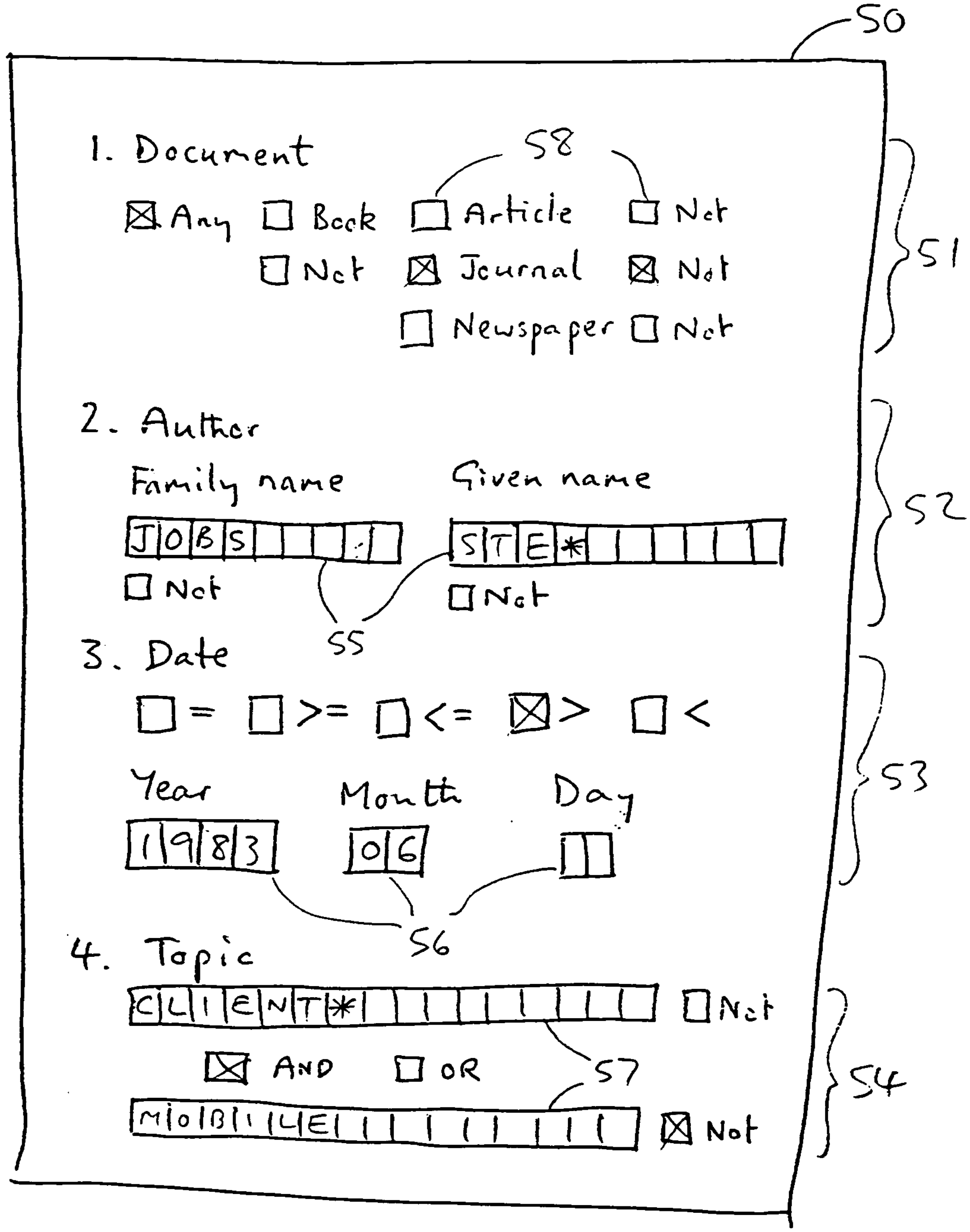

50
1. Document
58
Any
Book
Article
Not
Not
Journal
Not
Newspaper
Not
51
2. Author
Family name
Given name
JOBS
STE*
Not
Not
52
55
3. Date
=
>=
<=
>
<
Year
Month
Day
1983
06
53
56
4. Topic
CLIENT*
Not
AND
OR
57
54
MOBILE
Not

Fig. 5

SPARE

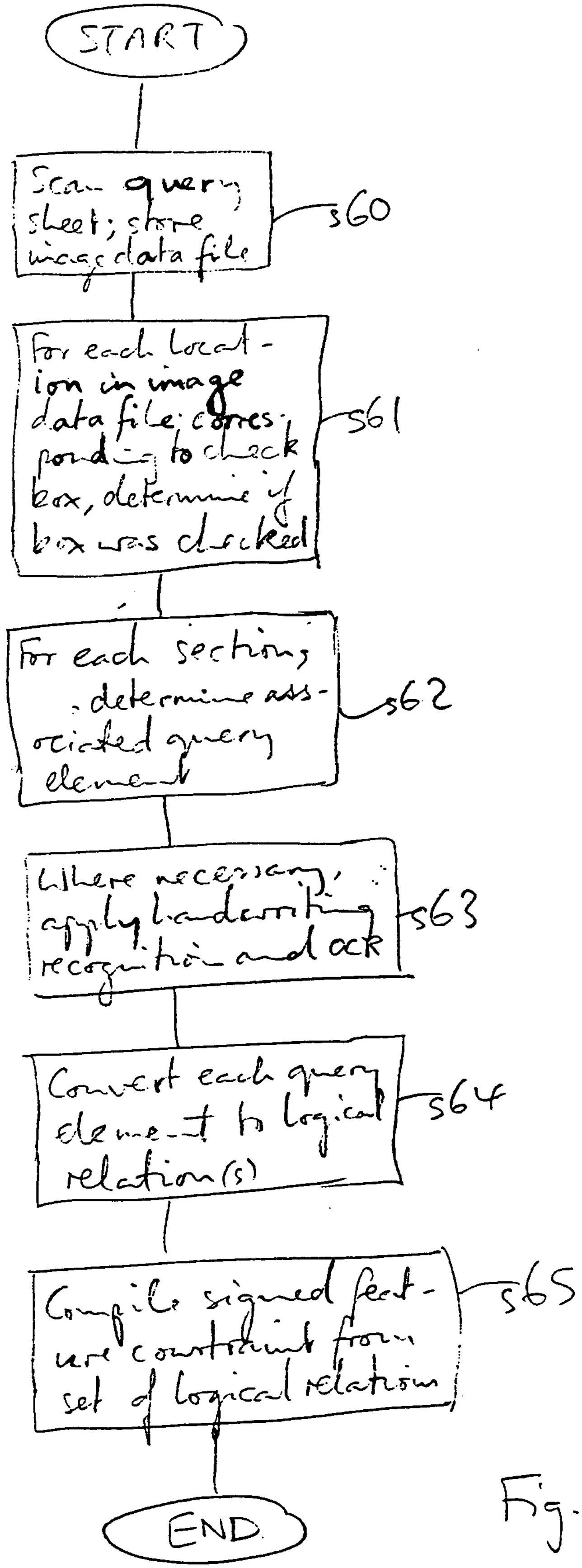

START
Scan query sheet; store image data file
s60
For each location in image data file corresponding to check box, determine if box was checked
s61
For each section, determine associated query element
s62
Where necessary, apply handwriting recognition and ocr
s63
Convert each query element to logical relation(s)
s64
Compile signed feature constraint from set of logical relations
s65
END

Fig. 6

SPARE

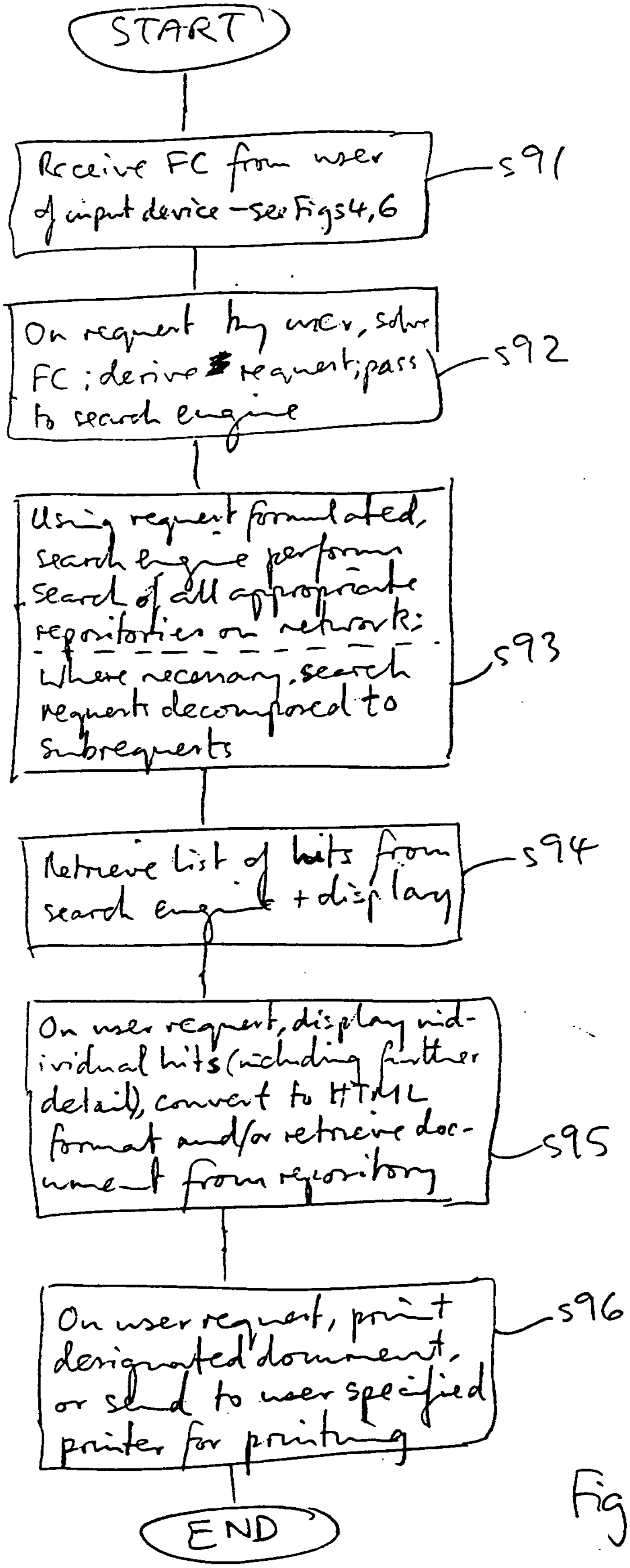
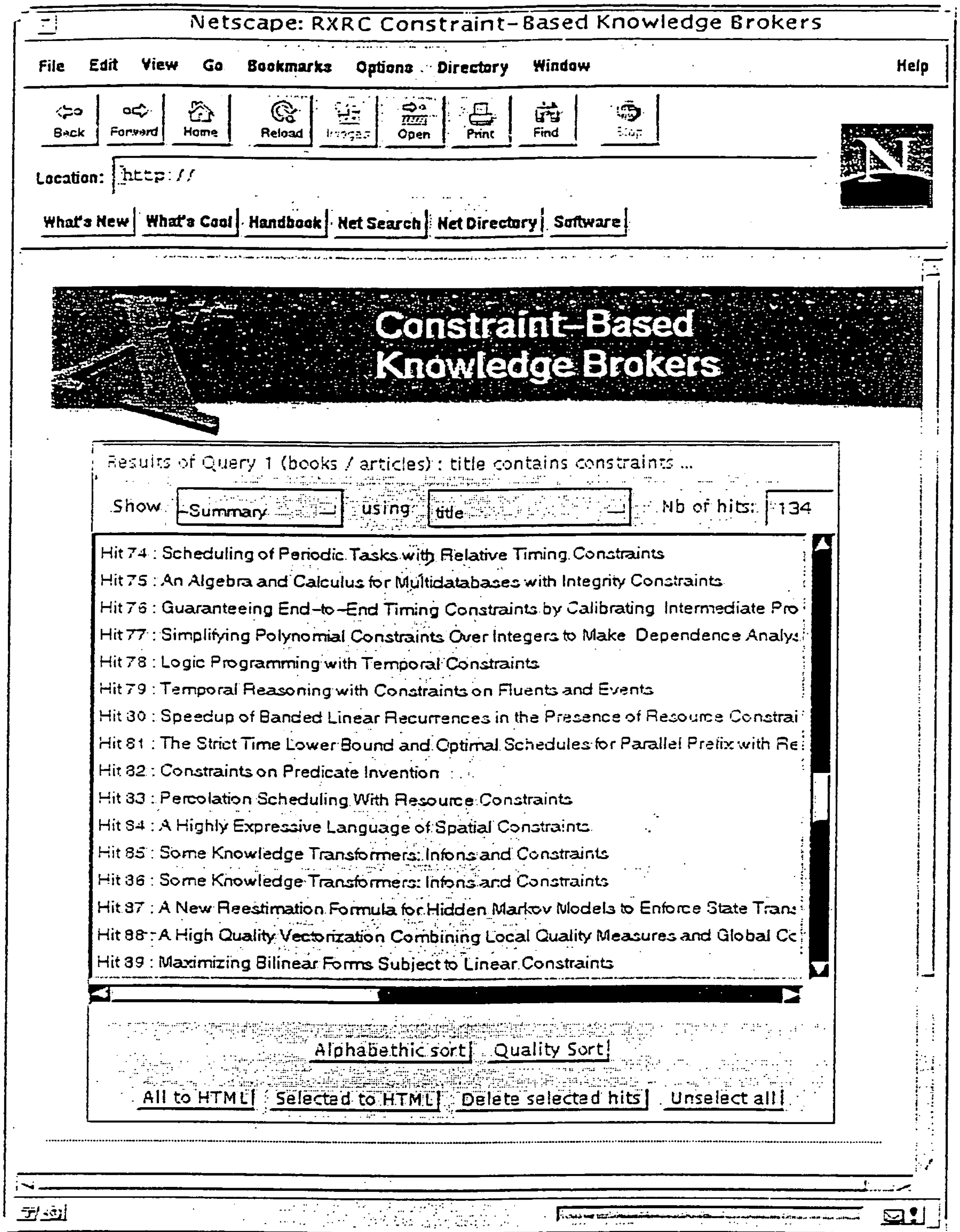

START
Receive FC from user of input device - see Figs 4,6
s91
On request by user, solve FC; derive request; pass to search engine
s92
Using request formulated, search engine performs search of all appropriate repositories on network;
Where necessary, search requests decomposed to subrequests
s93
Retrieve list of hits from search engine + display
s94
On user request, display individual hits (including further detail), convert to HTML format and/or retrieve document from repository
s95
On user request, print designated document, or send to user specified printer for printing
s96
END

Fig. 7



Netscape: RXRC Constraint-Based Knowledge Brokers
File Edit View Go Bookmarks Options Directory Window Help
Back Forward Home Reload Images Open Print Find Stop
Location: http://
What's New What's Cool Handbook Net Search Net Directory Software
Constraint-Based Knowledge Brokers
Results of Query 1 (books / articles) : title contains constraints ...
Show Summary using title Nb of hits: 134
Hit 74 : Scheduling of Periodic Tasks with Relative Timing Constraints
Hit 75 : An Algebra and Calculus for Multidatabases with Integrity Constraints
Hit 76 : Guaranteeing End-to-End Timing Constraints by Calibrating Intermediate Pro
Hit 77 : Simplifying Polynomial Constraints Over Integers to Make Dependence Analys
Hit 78 : Logic Programming with Temporal Constraints
Hit 79 : Temporal Reasoning with Constraints on Fluents and Events
Hit 80 : Speedup of Banded Linear Recurrences in the Presence of Resource Constrai
Hit 81 : The Strict Time Lower Bound and Optimal Schedules for Parallel Prefix with Re
Hit 82 : Constraints on Predicate Invention
Hit 83 : Percolation Scheduling With Resource Constraints
Hit 84 : A Highly Expressive Language of Spatial Constraints
Hit 85 : Some Knowledge Transformers: Infons and Constraints
Hit 86 : Some Knowledge Transformers: Infons and Constraints
Hit 87 : A New Reestimation Formula for Hidden Markov Models to Enforce State Trans
Hit 88 : A High Quality Vectorization Combining Local Quality Measures and Global Cc
Hit 89 : Maximizing Bilinear Forms Subject to Linear Constraints
Alphabethic sort Quality Sort
All to HTML Selected to HTML Delete selected hits Unselect all

Fig. 8

8/9

SPARE

Netscape: RXRC Constraint-Based Knowledge Brokers - Report

File Edit View Go Bookmarks Options Directory Window Help

Back Forward Home Reload Images Open Print Find Stop

Location: http://

What's New | What's Cool | Handbook | Net Search | Net Directory | Software

Constraint Based Knowledge Brokers Report

Information retrieved Wed, 4 Sep 96

Query: find all books where title contains constraint and date after 90 and title contains not internet

Reference	Hit-0001
Query	*Search in books servers*
author	Z. Zhang.
http_url	http://www.cs.umass.edu:80/Dienst/UI/2.0/Describe/ncstrl.umass_cs%2fUM-[illegible]
information_source	NCSTRL (Cornell Computer Science)
reference	UM-CS-1996-008
title	3D Reconstruction under Varying Constraints on Camera Geometry for Roboti[illegible]

Reference	Hit-0002
Query	*Search in books servers*
author	Masato Ishizaki
bibtype	InProceedings
crossref	coling-90
information_source	Glimpse server (Arizona)

Document: Done.

Fig. 9

SPARE

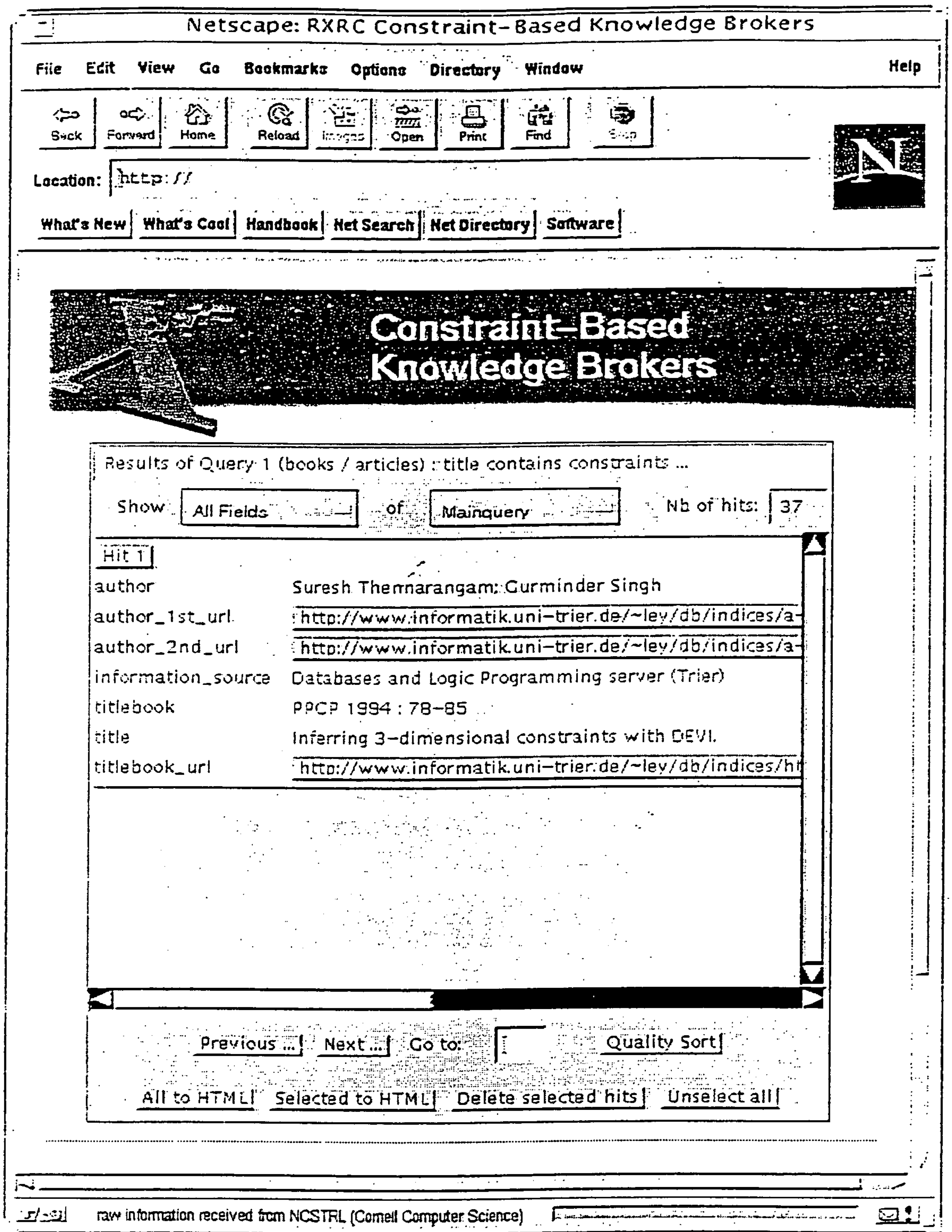

Netscape: RXRC Constraint-Based Knowledge Brokers
File Edit View Go Bookmarks Options Directory Window Help
Back Forward Home Reload Images Open Print Find Stop
Location: http://
What's New What's Cool Handbook Net Search Net Directory Software
Constraint-Based Knowledge Brokers
Results of Query 1 (books / articles) : title contains constraints ...
Show All Fields of Mainquery Nb of hits: 37
Hit 1
author Suresh Thennarangam; Gurminder Singh
author_1st_url http://www.informatik.uni-trier.de/~ley/db/indices/a-
author_2nd_url http://www.informatik.uni-trier.de/~ley/db/indices/a-
information_source Databases and Logic Programming server (Trier)
titlebook PPCP 1994 : 78-85
title Inferring 3-dimensional constraints with DEVI.
titlebook_url http://www.informatik.uni-trier.de/~ley/db/indices/ht
Previous ... Next ... Go to: Quality Sort
All to HTML Selected to HTML Delete selected hits Unselect all
raw information received from NCSTRL (Cornell Computer Science)

Fig. 10